|  |
| --- |
|  |
| Rozbudowa korporacyjnej bazy danych dla PIG |
|  |
|  |
| 24.05.2016 |
| 1.2 |

**Historia zmian**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Autor | Data | Wersja dokumentu | Wersja systemu | Uwagi |
| xxx | 20/7/2011 | 0.9 | 3.15.0 | Utworzenie dokumentu |
| xxx | 27/7/2011 | 1.0 | 3.15.0 | Korekta |
| xxx | 19/7/2012 | 1.1 | 3.16.0 | Aktualizacja dla Fazy 3 |
| xxx | 24/5/2016 | 1.2 | 3.21.0 | Realizacja umowy EZ-244-23/2015 |

**Spis treści:**

1. Wstęp 4

1.1. Cel 4

1.2. Powiązane dokumenty 4

1.3. Definicje 4

2. Budowanie aplikacji 6

2.1. Oprogramowanie 6

2.2. Źródła 6

2.3. Struktura aplikacji 8

2.4. Budowanie aplikacji 10

2.5. Utworzenie bazy danych 11

3. Dokumentacja projektów 15

# Wstęp

## Cel

Dokument ten zawiera dokumentację techniczną systemu Midas3. Zawiera informację o sposobie budowania systemu. Dokument przeznaczony jest dla projektantów / programistów odpowiedzialnych za utrzymanie systemu.

## Powiązane dokumenty

| **ID** | **Tytuł/opis** | **Nazwa pliku** |
| --- | --- | --- |
| 1 | Projekt architektury systemu Midas 3. | PIGxx-MIDAS-01 Projekt architektury ver. 1.0 |

## Definicje

|  |  |
| --- | --- |
| **Definicja** | **Opis** |
| Maven | Narzędzie budowania aplikacji na podstawie kodów źródłowych. |

# Budowanie aplikacji

## Oprogramowanie

### Użyte oprogramwanie systemowe

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Version** | **Platform** | **Address** |
| Oracle JDK | 1.6 | Linux / Windows | <http://www.oracle.com/technetwork/java/javase/downloads/index.html> |
| JBoss | 5.1.0 | All | <http://sourceforge.net/projects/jboss/files/JBoss/JBoss-5.1.0.GA> |

### Narzędzia programistyczne

|  |  |  |
| --- | --- | --- |
| **Name** | **Version** | **Address** |
| Eclipse IDE for Jave EE Developers | 3.6 | <http://eclipse.org/downloads/> |
| Maven | 2.2.1 | <http://maven.apache.org/download.html> |
| Tortoise SVN | 1.6.11 | <http://tortoisesvn.net/downloads> |
| Balsamiq Mockups | 1.8.12 | <http://balsamiq.com/download>  - download and install MockupsForDesktop.air |
| Firebug plugin for Firefox |  |  |
| HttpFox plugin for Firefox |  |  |
| PageSpeed plugin for Firefox |  |  |
| PageSpeed plugin for Chrome |  |  |

## Źródła

### Kod źródłowy

Kod źródłowy aplikacji jest przechowywany w repozytorium GIT na serwerze Pentacomp.

Struktura gałęzi GIT

|  |
| --- |
| **EtapI** |
| **EtapI** |
| **EtapIII** |
| **EtapIV** |

Gałęzie zawierają wszystkie kody źródłowe zaimplementowane w poszczególnych etapach. Aktualne kody źródłowe zawarte są na gałęzi EtapIV.

### Biblioteki

Biblioteki zewnętrzne i artefakty systemu (biblioteki jar i war wchodzące w skład aplikacji) przechowywane są w repozytorium na serwerze w Impaq.

Zdefiniowane są lokalne i zdalne repozytoria. W lokalne przechowywane są artefakty umieszczone przez programistów lub systemy automatycznego budowania aplikacji lub kopie artefaktów pobrane ze zdalnych repozytoriów. Zdalne repozytoria są udostępnione w Internecie i przechowują publicznie dostępne pliki bibliotek *open source*.

##### Lokalne repozytoria

|  |  |  |  |
| --- | --- | --- | --- |
| **Nazwa** | **Adres** | **Opis** | **Adres źródła** |
| PGI Releases group | <http://midas.test.impaq.com.pl:8080/artifactory/pgi-releases-group/> | Grupuje lokalne i zdalne repozytoria:   * PGI Releases only * public-jboss * jboss * java.net.m2 * repo1 |  |
| PGI Releases | <http://midas.test.impaq.com.pl:8080/artifactory/pgi-releases-local/> | PGI Releases only |  |
| PGI Snapshots | <http://midas.test.impaq.com.pl:8080/artifactory/pgi-snapshots-local/> | PGI Snapshots only |  |
| Plugins-releases | <http://midas.test.impaq.com.pl:8080/artifactory/plugins-releases-local/> | Plugins releases |  |
| OEM-releases | <http://midas.test.impaq.com.pl:8080/artifactory/oem-releases/> | OEM releases:   * com/esri/arcobjects/arcobjects-9.3.1:jar * com/oracle/ojdbc14/10.2.0.5.0/ojdbc14-10.2.0.5.0.jar * com/oracle/ojdbc6/11.1.0.7.0/ojdbc6-11.1.0.7.0 * jboss/jboss/5.1.0.GA/jboss-5.1.0.GA-jdk6.zip | OEM releases: |

##### Zdalne repozytoria

|  |  |  |  |
| --- | --- | --- | --- |
| **Nazwa** | **Id** | **Adres** | **Opis** |
| public-jboss | jboss-public-repository-group | <http://repository.jboss.org/nexus/content/groups/public-jboss/> | new jboss repository |
| jboss | jboss-maven | <http://repository.jboss.com/maven2> | old jboss repository |
| java.net.m2 | java.net-maven2-repository | <http://download.java.net/maven/2> | Java Maven 2 repository |
| repo1 | ibiblio | <http://repo1.maven.org/maven2> | Maven 2 default repository |
| google-gson | google-gson | <http://google-gson.googlecode.com/svn/mavenrepo> | Google gson repository |

## Struktura aplikacji

Struktura modułów aplikacji

* database
  + documentation

Generowanie dokumentacji bazy danych na podstawie schematu bazy danych

* + import

Skrypty umożliwiające import schematów bazy danych z plików .DMP

* + migration

Generowanie skryptów migracji z midas wersja 2 do 3

* gis-api

Api używane do usług GIS

* gis-web

Impelementacja kontrolki mapy GIS

* install

Skrypty budujące paczkę instalacyjną aplikacji łącznie z serwerem aplikacji

* midas-model

Model danych i usług

* karta złoża  
    
  Szablony JasperReports do generowania kart złoża
* midas-root

Model POM projektu

* midas-web

Aplikacja web

* pgi-root

Model POM organizacji PGI

* tools

Narzędzia

* test
  + gui-test

Automatyczne testy GUI

* + performance

Scenariusze testów wydajnościowych Selenium oraz konfiguracja testów wydajnościowych JMeter

## Budowanie aplikacji

Budowanie aplikacji jest możliwe za pomocą narzędzia Maven. Wspierana jest wersja 2.2.1. Opis instalacji narzędzia dostępny jest pod adresem <http://maven.apache.org/download.html#Installation>

Ustawienie Maven’a zapisane są w pliku *settings-impaq-pgi.xml* w katalogu źródeł *trunk.*

Plik konfiguracyjny należy skopowiać pod nazwą settings.xml do podkatalogu .m2 w katalogu domowym użytkownika (C:\Documents and Settings**\{użytkownik}** or /home/{**użytkownik}**.

Aby zbudować aplikację do uruchomienia na lokalnej maszynie należy wywołać standardową komendę budowania w głównym katalogu źródeł:  
mvn package -Dmaven.test.skip=true

mvn clean install

Komenda clean (usuwa lokalnie skoplikowane klasy, wygenerowane pliki) jest opcjonalna ale dobrym zwyczajem jest ją zawsze stosować.

Aby zbudować wersję przeznaczoną na któreś ze środowisk produkcyjnych / testowych np. midas.test.impaq.com.pl, testowe środowisko w PGI należy zbudować aplikację z włączonym odpowiednimi profilami np. pgitest,pgi – profile środowiska testowego w PGI. Przykładowe poleceni budowania wygląda następująco:

mvn clean install –Ppgitest

Powyższa komenda zbuduje aplikację na środowisko testowe w PIG.

### Profile

Główne profile zdefiniowane w settings.xml to:

* impaq-artifactory – dostarcza konfiguracji repozytoriów aplikacji, domyślnie włączony
* localhost – definiuje ustawienia lokalnego środowiska, domyślnie włączony
* midastest – definiuje ustawienia środowiska testowego w Impaq na serwerze midas.test.impaq.com.pl
* pgitest – definiuje ustawienia środowiska testowego w PGI na serwerze midas3.pgi.gov.pl

Dodatkowe profile dostępne w całej aplikacji:

* check – jeśli jest włączony przy budowaniu uruchamia kontrolę jakości kodu
* itest – jeśli jest właczony przy budowaniu uruchamia testy integracyjne

Dodatkowe profile zdefiniowane w projektach:

* dist (projekt install) – buduje paczkę dystrybucyjna aplikacji łącznie z serwerem aplikacji JBoss
* devinst (projekt install) – instaluje na lokalnej maszynie system rozpakowany z paczki dystrybucyjnej
* selenium (projekt test/gui-test) – uruchamia automatyczne testy GUI
* pgiprod (projekt gis-web i midas-web) – konfigurauje komponenty aplikacji dla środowiska PGI
* impaq (projekt gis-web i midas-web) – konfiguruje komponenty aplikacji dla środowiska Impaq

## Utworzenie bazy danych

### Wymagania

Lokalnie zainstalowane narzędzia Oracle: sqlplus oraz imp

### Import schematów i danych

Utworzenie bazy danych w środowisku testowym możliwe jest za pomocą automatycznych skryptów Maven zdefiniowanych w projekcie database/import.

|  |  |
| --- | --- |
|  | Podczas importu schematu PTERYT, NKI, MIDAS\_GIS zgłaszane są pojedyńcze ostrzeżenia,  przy imporcie schematu MIDAS2DEV2 jest ich dużo więcej. |

Odtworzenie automatyczne używa skryptów Maven.  
Skrypty zakładają następujące położenie i nazwy plików DMP:

<database.import.dump.dir>C:\opt\midas-dump</database.import.dump.dir>

<database.import.dump.pterytFile>$\{database.import.dump.dir\}/pteryt05102010.DMP</database.import.dump.pterytFile>

<database.import.dump.nkiFile>$\{database.import.dump.dir\}/nki05102010.DMP</database.import.dump.nkiFile>

<database.import.dump.midasFile>$\{database.import.dump.dir\}/midas2dev205102010.DMP</database.import.dump.midasFile>

<database.import.dump.gisFile>$\{database.import.dump.dir\}/midas\_gis\_new.DMP</database.import.dump.gisFile>

Konfiguracja połączenia do bazy danych określona jest za pomocą następujących parametrów:

<database.host>localhost</database.host>

<database.url>jdbc:oracle:thin:@${database.host}:1521:xe</database.url>

<database.username>xxx</database.username>

<database.password>xxx</database.password>

<database.admin.username>xxx</database.admin.username>

<database.admin.password>xxx</database.admin.password>

Wartości parametrów można nadpisać w pliku konfiguracyjnym Maven **settings.xml** znajdującym się w podkatalogu domowym użytkownika **C:\Documents and settings\${user}\.m2**

Odtworzenie wymaga wywołania w katalogu źródeł **trunk/database/import** następujących komend:

|  |  |
| --- | --- |
|  | W przypadku braku modyfikacji danych w schematach innych niż MIDAS3 import danych do pozostałych schematów można wykonać jednokrotnie. |

1. Usunięcie wszystkich istniejących użytkowników MIDAS\_GIS,MIDAS2DEV2,NKI,PTERYT oraz pozostałych
2. mvn clean install -Pdrop-others
3. Usunięcie istniejących użytkowników MIDAS3WWW,MIDAS3
4. mvn clean install -Pdrop
5. Utworzenie wszystkich użytkowników: MIDAS\_GIS,MIDAS2DEV2,NKI,PTERYT oraz pozostałych
6. mvn clean install -Pcreate-others

|  |  |
| --- | --- |
|  | **Jednokrotne** |

1. Utworzenie użytkowników MIDAS3WWW,MIDAS3
2. mvn clean install -Pcreate
3. Wykonanie importu użytkowników PTERYT,NKI z danych DMP
4. mvn clean install -Pimport-others

|  |  |
| --- | --- |
|  | **Jednokrotne** |

1. Wykonanie importu użytkownika MIDAS2DEV2 z danych DMP
2. mvn clean install -Pmidas2dev2

|  |  |
| --- | --- |
|  | **Opcjonalne lub jednokrotne** |

1. Wykonanie importu użytkownika MIDAS3 z danych DMP, nadanie praw oraz modyfikacja sekwencji
2. mvn clean install -Pimport,grants,set-seq
3. Aplikacja poprawki pakietów użytkownika MIDAS3
4. mvn clean install -Pfix

|  |  |
| --- | --- |
|  | **Po migracji danych** |

### Skrypty migracyjne

#### Struktura

Skrytpy zdefiniowane są w katalogu src/main/resources. Głównym plikiem konfiguracyjnym jest dbchangelog.xml włącząjący inne pliki definiujące migrację. Pliki korzystają z języka Liquibase o strukturze XML oraz skryptów SQL, PL/SQL.

#### Uruchomienie skryptów

Skrypty migracji generowane są za pomocą plugin’u [Liquibase](http://liquibase.org/) do narzędzia budowania Maven.

Po migracji wykonywane są dodatkowe skrypty PL SQL aktualizujące dane.

|  |  |
| --- | --- |
|  | Zakładane jest, że migracja rozpoczyna się po utworzeniu schematu **midas3** na podstawie schematu midas2 |

##### Pierwsza migracja

Pierwsza migracja na początkowej bazie danych bez schematów

1. Utworzenie **tag**'a o nazwie **dump** na bazie w wersji Midas 2 za pomocą komendy:

mvn liquibase:tag -Pliquibase -Dliquibase.tag=dump -Dliquibase.clearCheckSums=true

1. Wykonanie skryptów migracyjnych za pomocą komendy:

mvn clean install -Pliquibase

###### Migracja przyrostowa

Migracja przyrostowa skutkuje dodaniem tylko nowych zmian. Opiera się na sumach kontrolnych utworzonych przy wprowadzaniu zmian - tylko nowe zmiany zostaną dodane.

Wykonanie skryptów migracyjnych umożliwia komenda:

mvn clean install -Pliquibase

#### Utworzenie skryptów migracyjnych SQL

Możliwe jest utworzony skryptów migracyjnych SQL zawierających wszystkie polecenia SQL wykonywane w trakcie migracji. **Uwaga**: Zmiany trafiają tylko do pliku, nie są wykonywane zmiany w bazie danych

Utworzenie skryptów migracyjnych umożliwia komenda:

mvn clean liquibase:updateSQL -Pliquibase

Wynikiem działania jest wygenerowany plik znajdujacy się w podkatalogu target/liquibase o nazwie migrate.sql

Wszystkie dodatkowe skrypty bazodanowe znajdują się w katalogach:  
midas-model/src/main/resources/raporty/  
midas-model/src/main/resources/sql

Po zbudowaniu aplikacji i wgraniu wara na serwer aplikacyjny wszystkie tabele bazy danych zostaną utworzone.

# Dokumentacja projektów

Dokumentację projektów można wygenerować za pomocą narzędzia Maven. Służy temu polecenie:

mvn clean site

Wynikiem działania będzie katalog target/site zawierający szereg raportów w formacie HTML. Głównym plikiem raportu jest *index.html*.

Wygenerowana dokumentacja zawiera m.in. następujące raporty:

Project Summary – krótka informacja o projekci

Project Dependencies – zależności projektu do bibliotek

Source Repositories – informacja o adresie źródeł projektu

Checkstyle – raport zgodności z regułami CheckStyle.

Cobertura Test Coverage – raport z pokrycie kodu przez testy za pomocą Cobertura.

CPD Report - raport poziomu duplikacji kodu.

JavaDocs – dokumentacj JavaDoc API klas projektu.

PMD Report - raport zgodności z regułami PMD.

Source Xref - źródła w formacie HTML umożliwiające wygodne przeglądanie.

Surefire Report – raport z wyników testów jednostkowych.

Test JavaDocs – dokumentacja JavaDoc API dla klas testowych.

Test Source Xref – źródła testów w formacie HTML umożliwiające wygodne przeglądanie.